CHAPTER

6

WINDOWS XP SECURITY AND ACCESS CONTROLS

After reading this chapter and completing the exercises, you will be able to:

- Describe the Windows XP security model, and the key role of logon authentication
- Customize the logon process
- Discuss domain security concepts
- Understand the Local Computer Policy
- Enable and use auditing
- Encrypt NTFS files, folders, or drives using the Encrypted File System (EFS)
- Understand and implement Internet security

Windows XP, like Windows 2000 (and Windows NT), has been constructed to give a wide range of control over access to its resources. In fact, Windows XP is designed to check access permissions for every request before granting access to resources. In this chapter, we explore the details of the Windows XP security model, its logon process, and the ways in which the operating system associates security information with all objects under its control. Additionally, Windows XP efficiently subjects any user's or program's request for system resources to close scrutiny before allowing access to the requested resources.

The Windows XP Security Model

Windows XP Professional can establish local security when used as a standalone system or in a workgroup, or participate in **domain security** (either managed by a Windows .NET Server, Windows 2000 Server, Windows NT Server, or some other NOS). Before a user can access any Windows XP resource, he or she must log onto the system by supplying a valid user ID and **password**. A user who successfully logs on receives an **access token**. The access token includes information about the user's identity, any permissions specifically associated with the user's account name, and a complete list of all the groups to which the user belongs. A string of bits represents the token, which is attached to every **process** that the user initializes until that user logs off. In other words, each time a user runs a program, enters a system command, or accesses a resource, a copy of that user's access token accompanies the request.

Each time a user attempts to access a resource, the user's access token is compared with a list of permissions associated with the resource. This list is called an **access control list (ACL)**. The access control list is one of the more important attributes associated with any Windows XP resource. Whenever an object is requested, the ACL and the access token are carefully compared, and a request for the object is granted only when a match is found. Windows XP uses permission settings of Allow and Deny. An Allow setting enables a service for a user or group, whereas a Deny setting disables it. If neither Allow nor Deny is defined for a specific service for a user or group, it defaults to Deny.

If the system finds a match between the access token and the ACL, the request can proceed. If a requested resource is specifically denied in the ACL, or access to a service is not permitted, the request is denied. A match between the access token and the ACL is like finding a key that fits a particular lock. That is, the access token is like a ring of keys that you try in a lock one at a time until a match is found or until there are no more keys to try. Matches between an access token and the ACL can be a function of permissions associated with the individual user's account or permissions that derive from the user's membership in some particular local or global group. Whatever the source of the permissions, the user's request is allowed to proceed unhindered if a match is found.

Windows domain security is centered on Active Directory, the centralized database of security, configuration, and communication information maintained by domain controllers in a Windows network. Active Directory supports everything from authentication of domain users' accounts to accessing shared resources. Windows XP Professional, as a standalone system or as a member of a Windows NT 4.0 domain, does not use Active Directory, relying instead on the Registry and internal security systems to control user access. Note that if a Windows XP Professional system is used in a Windows NT domain, it must be authenticated by a domain controller. However, Windows XP Professional participates in Active Directory when it is used as a client in a Windows 2000 or Windows .NET domain network. All of the information about the domain and all of the resources shared by the network are managed by Active Directory. Windows XP Professional uses Active Directory to gain access to the domain network.

Logon Authentication

Windows XP logon is mandatory to gain access to the system and to applications and resources. As mentioned in Chapter 5, "Users, Groups, Profiles, and Policies," there are two types of logon available on Windows XP: classic and Windows Welcome. When the Windows XP system is a member of a domain, only the classic method can be used. But, when it is a standalone system or a member of a workgroup, either classic or Windows Welcome can be used.

The logon process typically has two components: identification and authentication. **Identification** requires that a user supply a valid account name (and in a domain environment, the name of the domain to which that **user account** belongs). **Authentication** means that a user must use some method to verify his or her identity. By default in Windows XP, possession of the proper password for an account constitutes authentication, although Windows XP also supports third-party authentication add-ins, including biometric systems that check fingerprints or perform retinal scans and smart card systems that require physical possession of a unique electronic keycard to prove a user's identity. If an account does not have a password assigned to it, the authentication process is skipped.

Most typical Windows XP systems rely solely on passwords for authentication, so using hard-to-guess passwords is an important aspect of good system security. Good passwords generally include both uppercase and lowercase letters, as well as numbers, for example, Ag00dPA55w0Rd. By creating passwords such as this, it becomes impossible for programs that attempt system break-ins to gain access through using dictionary lists to search for valid passwords. It is important, however, to make passwords easy to remember so users don't write them down, thus creating an additional security risk.

When a user successfully logs onto a Windows XP machine, the security subsystem within the Executive Services layer creates an access token for that user. The access token includes all security information pertaining to that user, including the user's **security ID (SID)** and SIDs for each of the groups to which the user belongs and the associated rights and privileges. Indirectly, through the user rights policy, this collection of SIDs informs the system of the user's rights. An access token includes the following components:

- The unique SID for the account
- A list of groups to which the user belongs
- A list of rights and privileges associated with the specific user's account

Access to the system is allowed only after the user receives the access token. Each access token is created for one-time use during the logon process. Once constructed, the access token is attached to the user's **shell** process, which defines the environment inside which the user executes programs or spawns other processes. (The default shell process for Windows XP is Windows Explorer. It defines the desktop, Start menu, taskbar, and other elements of the default user interface. Alternate shells can be employed from third parties,

or even the Windows NT 3.51 Program Manager can be used.) As far as Windows XP is concerned, a process is a computer program design for some specific function. The term process is synonymous with program. All activities within the user mode and kernel mode are performed by a process. Each process is launched using the access token of its parent (i.e., the process that caused it to be launched). When a user launches a process manually, he does so through the shell process, usually Windows Explorer, and it is the access token of the shell process that is inherited by the new process.

Objects

In Windows XP, access to individual resources is controlled at the **object** level. Each object hosts its own access control list that defines which users and groups have access permissions and exactly what type of access they are granted (read, write, print, delete, add, list, etc.). Everything within the Windows XP environment is an object, this includes files, folders, processes, user accounts, printers, computers, etc. Requests for resources, therefore, translate into requests for objects. An individual object is identified by its type, which defines its permitted range of contents and the kinds of operations (called services) that may be performed upon it. Any individual object is an instance of its type and consists of data and a list of services that can be used to create, manipulate, control, and share the data it contains.

Windows XP is able to control access not only at the object level, but also to control which services defined for the object's type a particular security token is allowed to perform or request. All objects are logically subdivided into three parts: a type identifier, a list of services or functions, and a list of named attributes that may or may not have associated data items, called values.

When defining an object, its type describes the kind of entity it is. For example, an object's type may be file, directory, printer, or network share. An object's services define how the object can be manipulated; for example, possible services for a directory object are Read, Write, and Delete. An object's attributes are its named characteristics, such as the file's name and size, whether it is read-only or hidden, and data created for an object whose type is file. The value for these attributes is their content, such as the actual name of the file, selected, not selected, 142,302 bytes, and 10/3/99 04:23:34 PM, respectively.

Remember, access or permission to use an object is determined on the basis of the entire object and also for each of the services defined for that object. For example, a user can have access to read a file, such as an email program's executable (.exe) file, but not to edit or delete it. Thus, the user can have permission to access the object in general, but there may be more specific controls about what services they can request in connection with that access.

Windows XP automatically grants the Everyone group Full Control to the object whenever a new object or share is created. Thus, you must implement restrictions on new objects and new shares. In other words, Windows XP allows everyone to access to new objects by default.

Access Control

The classic Windows XP logon process is initiated through the attention sequence (the Ctrl+Alt+Delete keystroke combination, known to many DOS users as the "three-finger salute"). This attention sequence initiates a hardware interrupt that cannot be "faked" by a program and brings up a logon procedure dialog box that is stored in a protected area of memory, thus securing the system from attack through an unauthorized logon.

This combination of characteristics for the logon authentication procedure is the key to the entire Windows XP security scheme, because all other security features are based upon the level of authority granted a user who has successfully logged on. The Windows XP security structure requires a user to logon to a computer with a valid username and password. Without this step, nothing more can be accomplished in the Windows XP environment.

The Windows XP logon procedure provides security through the use of the following:

- *Mandatory logon*—The user must logon to access the computer.
- *Restricted user mode*—Until a successful logon takes place, all user-mode privileges are suspended. Among other things, this means that the user cannot launch applications, access resources, or perform any action or operation on the system.
- *Physical logon*—The structure of the logon sequence ensures that the logon occurs from the local keyboard, rather than from some other internal or external source. This is because the attention sequence initiates a hardware interrupt that accepts input only from the local keyboard.
- *User profiles*—Windows XP allows each user that logs on to a particular machine to save user preferences and environment setting, called a user profile. Each user can have a set of specific preferences restored at logon or can be supplied with a mandatory or default set, depending on how the system is configured. A user profile that is configured to follow a user throughout a network is called a roaming profile. (Profiles are covered in detail in Chapter 5.)

Customizing the Logon Process

A system administrator can alter the default logon process appearance and function using WinLogon. The **WinLogon** process produces the logon dialog box, where user name, password, and domain are selected. WinLogon also controls automated logon, warning text, the display of the Shutdown button, and the display of the last user to log onto the system. WinLogon operates in the user mode portion of the Windows XP system architecture and communicates with the Security Reference Monitor and SAM (Security Accounts Manager) database in the kernel's Executive Services to authenticate users and

launch their environment shell with an attached user-specific access token. The WinLogon process can be customized to display some or all of the following characteristics:

- Retain or disable the last logon name entered
- Add a logon security warning
- Change the default shell
- Enable/Disable the WinLogon Shutdown button
- Enable automated logon

Most of these characteristics can be altered through the Local Security Policy (accessed from Start|Control Panel|Switch to Classic View|Administrative Tools|Local Security Policy; see Chapter 5). All of these characteristics can be controlled through the Registry through the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\ key (see Figure 6-1). However, Microsoft recommends using the Local Security Policy interface to alter these items instead of the Registry when possible.

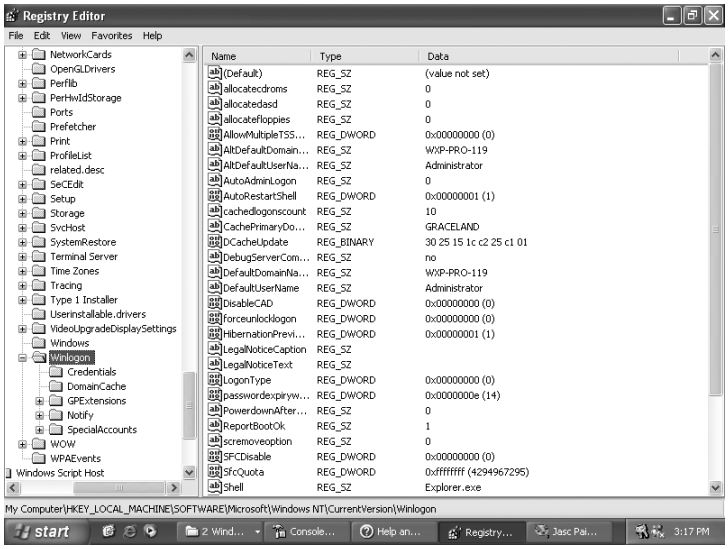

Figure 6-1 The Winlogon key viewed through Regedit

Several possible configuration changes for the Windows XP logon process are detailed in the following sections.

Disabling the Default Username

By default, the logon window displays the name of the last user to log on. If the same user consistently logs on to a single machine, displaying the logon name is convenient; however, for shared or public-access machines, this provides a key piece of information that someone could use to break into your system. It is possible to change the default by altering the value of its associated Registry key (DontDisplayLastUserName) or Local Security Policy value. Another common form of attack is a dictionary attack, which involves supplying the contents of a dictionary, one word at a time, as a logon password. Avoiding such systematic break-in attempts explains why it's a good idea to limit the number of failed logon attempts though the Lockout Policy.

Disabling this option presents a blank username field at the logon prompt. Note that the related value and its corresponding assignment do not occur in the Registry by default. The value is named DontDisplayLastUserName, and it is of type "String," where a value assignment of 1 disables the name display and a value of 0 (zero) enables it. This control appears by default in the Local Computer Policy. As noted, it is recommended that you use the Local Computer Policy to manage this feature rather than edit the Registry. (Try Hands-on Project 6-2 to disable the display of the user name of the last successful logon.)

Adding a Security Warning Message

Depending on your organization's security policy, you might be legally obligated to add a warning message that appears before the logon prompt is displayed. U.S. law states that if you want to be able to prosecute individuals for unauthorized entry to or use of a system, you must warn all users that usage is monitored, unauthorized access is forbidden, and that unauthorized users might be liable for prosecution.

Two Registry or Local Security Policy values are involved in this effort:

- *LegalNoticeCaption*—Puts a label on title bar of the legal notice window that appears during logon. This field works best with 30 characters of text or less.
- *LegalNoticeText*—Contains text information that provides the details of the warning to be issued to system users. This field may be up to 65,535 characters long, but most warning messages do not exceed 1,000 characters in length.

After this feature has been activated and configured, a warning message appears each time a user enters the Windows XP attention sequence. This message requires the user's acknowledgment by clicking OK before the logon window is displayed. (Try Hands-on Project 6-3 to add a Legal Notice to your logon.)

Changing the Shell

The default shell (the application launched by WinLogon after a successful logon) is Windows Explorer. You can change the shell to a custom or third-party application depending on the needs or security policy of your organization. For example, you could

change the shell to use the Program Manager familiar to NT 3.51 and Windows 3.x users. To make this change, you change the Shell value in the Winlogon key from EXPLORER.EXE to PROGMAN.EXE.

If you change from the Windows Explorer shell to the Program Manager shell, your system will lose its onscreen taskbar and you will no longer be able to use the Start menu to launch programs from your desktop. This is why most organizations use the default shell.

Disabling the Shutdown button

By default, the Windows XP logon window includes a Shutdown button. However, in an environment in which users have access to the keyboard and mouse on a Windows XP machine, this option has the potential for unwanted system shutdowns, regardless of whether the system is a Windows XP Professional or server machine. Fortunately, this option can be disabled. It should be noted, however, that if the user still has access to the physical power switch on the computer, disabling this option might cause more headaches than it solves. A system that has been shut down or rebooted through the operating system has a much higher chance of coming back up successfully than one that was just powered off. Note that by default, this button is enabled for Windows XP Professional machines, but disabled for Windows Server machines.

The value named ShutdownWithoutLogon is the one you'll need to edit in either the Registry or the Local Security Policy. It's enabled (set to the value 1) by default. To disable this button, change its value assignment to 0 (zero); to re-enable it, reset its value to 1. When the button is disabled, it still appears in the WinLogon window, but it's grayed-out and unusable.

For laptops or other advanced computers with automatic shutdown capabilities, an additional button labeled "Shutdown and Power Off" appears. Similar machines might also support a sleep mode, in which all processing is suspended and all power turned off, except to the computer's RAM. In that case, Sleep also shows up as a shutdown option. This particular setting permits users to eliminate most of a computer's power consumption, yet be ready to resume activity at the push of a single button or movement of the mouse. If the machine warrants such settings, users will find related Registry values in their WinLogon key settings that help them control how these functions are handled.

Be aware that leaving the Shutdown button enabled means that anyone with access to the keyboard can enter the Windows attention sequence and shut down the local machine.

Automating Logons

Some special- or limited-used Windows XP machines (for example, airport kiosks or hotel information stations) may need to be always available and always logged into a low-security account for access to some dedicated application. Although the logon process cannot be bypassed, the values for username and password can be coded into the Registry to automate logons. This normally is of interest only when installing machines for public use, such as for information centers, kiosks, museum guides, or other situations in which a computer is used to provide information to the public. In such cases, it's important to have computer and user policies to prevent Windows XP-savvy users from attempting to break out of the public application and exploring other, less public aspects of the system (or worse, of the network to which it may be attached).

To set up an automated logon, the following Registry value entries must be defined and set within the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon key:

- *DefaultDomainName*—Defines the name of the domain to log into (needed only when logging into a networked machine that's part of a domain).
- *DefaultUserName*—Defines the default logon account name.
- *DefaultPassword*—Defines the password associated with the default account name. This value is not present by default. When auto logon is disabled, delete this value, because it stores the password in plain text.
- *AutoAdminLogon*—Instructs the machine to log itself on immediately following each boot-up. A value of 1 automatically logs on using the credentials from the other three values in this list. A value of 0 (zero) disables the auto-logon feature.

Automated logons create a situation in which the computer automatically makes itself available to users without requiring an account name or a password. It is essential, therefore, that this capability be exercised *only* when security is not a concern (if a machine hosts only a single application and is not connected to the network) or if access to the equipment is otherwise controlled.

Automatic Account Lockout

Automatic account lockout disables a user account if a predetermined number of failed logon attempts occur within a specified time limit. This feature is intended to prevent intrusion by unauthorized users attempting to gain access by guessing a password or launching a dictionary attack. The default setting in Windows XP is to allow an unlimited number of failed access attempts to a user account without locking out that account. However, this is not recommended when there is even a remote chance unauthorized people can gain physical access to logon consoles. The Windows XP account lockout feature is discussed in Chapter 5.

Domain Security Concepts and Systems

A **domain** is a collection of computers with centrally managed security and activities. A domain offers increased security, centralized control, and broader access to resources than any other computer system configuration. Security policies are domain-wide controls that specify password requirements, account lockout settings, auditing, user rights, security options, and more.

Domain Security Overview

Domain security is the control of user accounts, group memberships, and resource access for all members of a network instead of for only a single computer. All of the information about user accounts, group memberships, group policies, and access controls for resources are contained in the Active Directory, a database maintained by one or more domain controllers. A **domain controller** is a Windows 2000 or Windows .NET Server system with the Active Directory support services installed and configured.

Kerberos and Authentication Services

Authentication takes place in a Windows domain network under two conditions: interactive logon and network authentication. Interactive logon occurs when you press the attention sequence, then enter your user name and password. If you log onto a local system, such as a standalone Windows XP Professional system, all authentication is performed by the local security subsystem. If you are logging onto a domain, the local security subsystem communicates with a domain controller using **Kerberos** v5—an authentication encryption protocol—to protect your logon credentials.

A **network authentication** occurs when you attempt to connect to or access resources from some other member of the domain network. Network authentication is used to prove that you are a valid member of the domain, your user account is properly authenticated, and that you have access permissions to perform the requested action. The communications that occur during network authentication are protected by one of several methods, including:

- Kerberos v5
- Secure Socket Layer/Transport Layer Security (SSL/TLS)
- NTLM (NT LAN Manager) authentication for compatibility with Windows NT 4.0

The authentication protection method is determined by either the communication mechanism (such as IIS or standard network connection) or the settings in the Local Security Policy. Only one of these options is used, but both can be "active" at one time. This ensures that the server can actively respond as the client requests or uses one or the other. The server responds with the same authentication scheme requested by the client.

Kerberos Version 5 Authentication

Windows XP uses Kerberos version 5 as the primary protocol for authentication security. The system uses this protocol to verify the identity of both the client (user) and server (network service or application) upon each resource access. This is known as mutual authentication. It protects the server from unauthorized clients and prevents the user from accessing the wrong or spoofed servers. (A spoofed server is one that is programmed to appear to be a particular server when it is another; this is most common on the Internet when an attacker is trying to gain access to credit card information.)

The Kerberos version 5 authentication system was designed to allow two parties to exchange private information across an open network, such as the Internet. Kerberos version 5 assigns a unique key, called a ticket, to each user that logs on to the network. This unique ticket is then embedded in messages to identify the sender of the message to the message's recipient.

The Kerberos process is completely invisible to the user. For more information on Kerberos, consult the *Microsoft Windows XP Professional Resource Kit* from Microsoft Press.

Secure Socket Layer/Transport Layer Security

Secure Socket Layer/Transport Layer Security (SSL/TLS) is an authentication scheme often used by Web-based applications and is supported on Windows XP through IIS (Internet Information Server). SSL functions by issuing an identity **certificate** to both the client and server. A third-party Certificate Authority that both the client and server have chosen to trust, such as VeriSign (*http://www.verisign.com*), issues these certificates. When a resource request is made, the client sends its certificate to the server. The server verifies the validity of the client certificate, then sends its own certificate to the client along with an encryption key. The client verifies the validity of the server certificate, then uses the encryption key to initiate a communication session with the server. This encrypted communication link is used for all future communications during this session. Once the session is terminated, the link must be rebuilt by starting over with the client sending its certificate to the server.

For more information on SSL, consult the *Microsoft Windows .NET Server Resource Kit* and the *IIS Resource Kit*.

NTLM

NTLM (NT LAN Manager) authentication is the mechanism used by Windows NT 4.0. Windows XP supports this authentication method solely for backward compatibility with Windows NT Servers and Windows NT Workstation clients. NTLM functions by using a static encryption level (40-bit or 128-bit) to encrypt traffic between a client and server. NTLM is significantly less secure than Kerberos version 5.

For more information on NTLM, consult the *Microsoft Windows .NET Server Resource Kit* or the *Windows NT 4.0 Server Resource Kit*.

Local Computer Policy

Another security control built into Windows XP is the **Local Computer Policy**. This policy is a combination of controls that in Windows NT existed only in the Registry, through system policies, or as Control Panel applet controls. Sometimes the local computer policy is called a software policy or an environmental policy or even a Windows XP policy. No matter what name is actually used, the local computer policy is simply the local system's group policy (see Chapter 5). The effective policy is the result of the combination of all group policies applicable to the system.

In a Windows XP domain network environment, the local computer policy is controlled on a domain basis on a Windows domain controller. This control is based on site, domain, and organizational unit group policies. On a Windows XP Professional system, you can manually launch the MMC and add in the Global Policy snap-in to manage or change the local computer policy. You cannot manage domain policies from a Windows XP Professional machine.

There is also a Local Group Policy tool (called the Local Security Policy) accessed from Administrative Tools in the Control Panel. However, this tool is limited to only the Computer Configuration, Windows Settings, Security Settings subsection of the full GPO. The remainder of this chapter takes the perspective of working from the MMC snap-in instead of the Administrative Tools utility.

The contents of the local computer policy are determined during installation based on system configuration, existing devices, and selected options and components. Custom policies can be created through the use of .adm files (administrative templates), such as those used by the Windows NT 4.0 System Policy Editor. Such files from Windows NT 4.0 can be used with Windows XP with some caveats. When you open and edit the local group policy, you are working with the system.adm file. The .adm files used by the Group Policy editor reside in the \inf subfolder of the main Windows XP directory. Third-party software vendors can use custom .adm files to add additional environmental controls based on their software or services. To learn about creating custom .adm files, see the *Microsoft Windows .NET Server Resource Kit*.

The local computer policy is divided into two sections (see Figure 6-2): Computer Configuration and User Configuration. The Computer Configuration section contains controls that focus on the computer, such as hardware and software settings. The User Configuration section contains controls that focus on the user and the user environment, such as permissions and desktop settings.

Because the local computer policy contains over 300 individual controls, you should take the time to peruse the entire collection level by level. (Try Hands-on Project 6-1 to view the Local Computer Policy.)

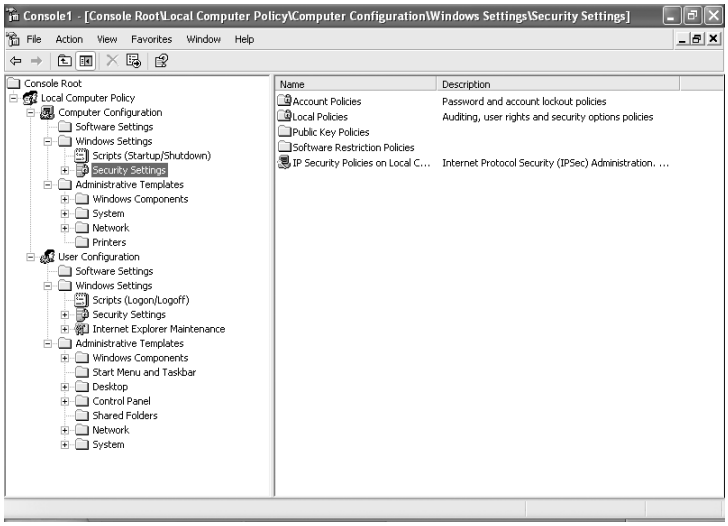

Figure 6-2 MMC with Group Policy snap-in displaying Local Computer Policy with Security Settings selected on a Windows XP Professional system

Computer Configuration

The Computer Configuration section of the local computer policy contains three subnodes or folders: Software Settings, Windows Settings, and Administrative Templates. Software Settings is empty by default; most third-party add-in application settings appear in this node. The Windows Settings folder contains two items: Scripts and Security Settings. The Scripts item allows you to define one or more scripts to be automatically executed at system startup or shutdown. The Security Settings is a container for settings for Account Policies, Local Policies (Audit, User Rights, and Security Options), Public Key Policies, and IP Security Policies. Account Policies and Local Policies were discussed in Chapter 5; Public Key Policies and IP Security Policies are discussed in the upcoming sections. The Administrative Templates folder contains a multilevel collection of computer-related controls (see later this chapter).

Public Key Policies

There are three purposes for using the **public key policies** control features: to offer additional controls over the Encrypted File System (EFS), to enable the issuing of certificates, and to allow you to establish trust in a certificate authority. The details on how to use this section of the local computer policy are hazy in the initial Windows XP Professional documentation. Consult the *Microsoft Windows .NET Server Resource Kit* for complete details. (Try Hands-on Project 6-5 to encrypt files with EFS.)

IP Security Policies

IP Security (IPSec) is a security measure added to TCP/IP to protect communications between two systems using that protocol. IPSec negotiates a secure encrypted communications link between a client and server through public and private encryption key management. IPSec can be used over a RAS or WAN link (through L2TP) or within a LAN. In either case, IPSec creates a secured point-to-point link between two systems. IPSec offers protection against a wide range of security problems, including: eavesdropping, data modification, identity spoofing, password attacks, denial of service attacks, man-in-the-middle attacks, compromised security key attacks, sniffer attacks, and application layer attacks.

IPSec is configured and enabled on each system through the Advanced TCP/IP Settings dialog box's Option tab. IPSec is enabled by default, but it is not configured. To configure IPSec, you must select one of the IPSec policies defined for your system/network. These policies are defined through the Group Policy.

IPSec can be used in one of two modes: transport or tunneling. In transport mode, an IPSec link can be established between any two systems on the network. In tunneling mode, an IPSec link can be established only between two specific systems. In other words, transport mode allows connections between any two systems on a network that are configured to use IPSec, whereas tunneling mode can be used only between two distinct partners. IPSec tunneling mode is often used to establish secure pathways between systems that often communicate critical or sensitive data, such as routers, gateways, or domain controllers.

The IP Security policies govern how a system communicates through TCP/IP, based on your defined security needs. Windows XP includes three predefined IPSec policies; however, you can create and manage your own custom IPSec policies. None of the predefined IPSec polices are enabled or assigned by default. For information on creating custom IPSec policies, consult the *Microsoft Windows .NET Server Resource Kit*.

The three predefined IPSec policies are Client (Respond Only), Server (Request Security), and Secure Server (Require Security). The Client (Respond Only) policy is for systems that do not require secure communications at all times. This policy initiates a secure communications link only when another system requests it. This policy does not initiate secure communications by default. The Server (Request Security) policy is for systems that need to use secure communications most of the time. This policy always requests that communications be secured, but allows unsecured communications to occur if IPSec is not available on the other system. The Secure Server (Require Security) policy is for systems that require secure communications at all times. This policy allows communications only if the remote system offers IPSec. Each of these policies can be modified through their Properties dialog box. However, Microsoft recommends creating new policies instead of modifying the default policies.

For an IPSec link to be established—whether as a VPN L2TP link over the Internet or simply a secured connection between two systems on the same LAN—a common

authentication method must be defined and available on both systems acting as the endpoints of the secure pipeline. Multiple authentication methods can be defined on a single system, but without a common method or rule, communication will not take place.

IPSec supports three types of authentication methods: Kerberos v5, public key certificate, and pre-shared key. Kerberos v5 is the default and preferred method of authentication and can be used by any client within the domain to establish a secured IPSec link. Public key certificate authentication can be used when systems not running Kerberos must be linked or when the systems are not members of the same domain. Public key certificate authentication is often used when linking across the Internet and over remote access links. Windows XP and Windows 2000 both support X.509 Version 3 certificates. This type of authentication requires at least one commonly trusted certificate authority (CA) between the two connecting systems. Without a commonly trusted CA, the link will not be established. Pre-shared key is supported by Windows XP IPSec, but it is seen as the least secure authentication option. It simply requires that each system use a common predetermined key, (a key is a string of characters, like a password). This pre-shared key is used to protect the initial authentication of a IPSec link. Pre-shared key authentication can be used by nearly any computer system that supports IPSec, thus not limiting connections to systems supporting Kerberos v5, Windows 2000/XP, or supporting a common public key certificate type or CA.

Administrative Templates

Administrative templates offer controls on a wide range of environmental functions and features. The Administrative Templates are Registry-based Group Policy information. In other words, Administrative Templates are used to overwrite the Registry of a client or server system to force compliance with the Group Policy. The Administrative Templates folder in the Computer Configuration node of the Local Computer Policy snap-in contains folders and subfolders with specific control items focused on a single aspect of the computer or environmental function. The controls available through the Administrative Templates folder include:

- Controlling security and software updates for Internet Explorer
- Controlling access and use of the Task Scheduler and Windows Installer
- Controlling logon security features and operations
- Controlling disk quotas
- Managing how group policies are processed
- Managing system file protection
- Managing offline access of network resources
- Controlling printer use and function

User Configuration

The User Configuration portion of the Local Computer Policy is structured in much the same way as the Computer Configuration portion. The User Configuration folder is also divided into three subfolders: Software Settings, Windows Settings, and Administrative Templates. Software Settings is empty by default. Any user-specific Microsoft or third-party software settings appear in this folder. The Windows Settings folder contains three items: Internet Explorer (IE), Scripts, and Security Settings. The Internet Explorer section is used to control user-specific activities of IE, such as browser interface appearance, connection methods, links, and security zones. The Scripts item allows you to define one or more scripts to automatically execute at user logon or logoff. Security Settings is initially empty, but can become a container for user-specific (rather than computer-specific) security controls. The Administrative Templates folder contains a multilevel collection of user-specific functional and environmental Registry-based controls. These controls are for the local computer only; if the computer is a member of a domain, some of these controls may be overwritten by a group, a computer, or by an organizational unit policy from the domain. The items contained in the User Configuration's Administrative Templates section include:

- Internet Explorer configuration, interface, features, and functions controls
- Windows Explorer management (interface, available commands, features)
- MMC management
- Task Scheduler and Windows Installer controls
- Start menu and Taskbar features management
- Desktop environment management
- Control Panel applet management
- Offline network access control
- Network connection management
- Logon and logoff script management
- Group Policy application

For more information on any control in the Local Computer Policy, open its Properties dialog box and view the Explain tab (see Figure 6-3). Try Hands-on Project 6.6.

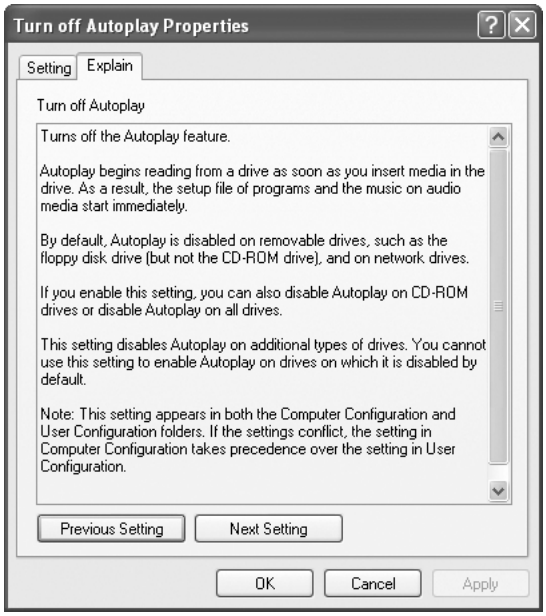

Figure 6-3 The Explain tab of a Local Computer Policy control dialog box

The Policy tab on the Properties dialog box for each control offers three settings:

- *Not Configured*—The default for all controls; does not change the existing setting of this control.
- *Enabled*—Enables the function or restriction of this control.
- *Disabled*—Disables the function or restriction of this control.

By carefully reading the materials on the Explain tab, you'll be able to understand which of the three settings for each control makes the most sense for the action you want to enforce or allow. In some cases, selecting the Enable control reveals additional controls, such as selection lists, numerical entry fields, or text-entry fields that provide the additional settings required by some controls. For example, the timeout settings require you not only to enable the control but also to define a time period in minutes or seconds for that timeout period.

Secedit

Secedit (Security Editor) is the command-line version of the Security Configuration and Analysis tool. Secedit is used to analyze, configure, export, and validate security based on a security template. A security template is a predefined group policy file with specific levels of security. The four functions of secedit each have their own specific parameters and syntax.

There are seven predefined security templates included with Windows XP Professional. They are:

- *compatws*—Configures security of a client to be the compatible with most non-certified applications.
- *hisecdc*—Configures the security of a domain controller to be at highest level possible.
- *hisecws*—Configures the security of a client to be at highest level possible.
- *rootsec*—Applies default root permissions to the boot partition (the one where the OS primarily resides) and to all child objects within the root.
- *securedc*—Configures the security of a domain controller to be at a moderate level.
- *securews*—Configures the security of a client to be at a moderate level.
- *setup security*—Configures the security of a system to the original default state after a typical installation.

The parameters and syntax of secedit are as follows:

```
Secedit /analyze /db FileName [/cfg FileName] [/log FileName]
       [/quiet]

Secedit /configure /db FileName [/cfg FileName] [/overwrite]
       [/areas area1 area2] [/log FileName] [/quiet]

Secedit /export [/mergedpolicy] [/db FileName] [/cfg FileName]
       [/areas area1 area2] [/log FileName] [/quiet]

Secedit /validate FileName
```

- *analyze*—Used to compare the current configuration of a system against a predefined security template.
- *db FileName*—Defines the path and filename of the security database. If no database file currently exists, the */cfg* parameter must be used in conjunction with */db*.
- *cfg FileName*—Defines the path and filename of the security template that will be imported into the database. This parameter can only be used with */db*. If */cfg* is not used, the action is performed against the security template already loaded into the database.
- *log FileName*—Defines the path and filename of a log file, if this is not specified a default log file is used.
- *quiet*—Suppresses all screen and log output.
- *configure*—Used to forcibly apply a security template to a system.

- *overwrite*—Indicates whether the imported security template should overwrite any template already stored in the database. In not specified, the imported security template will overwrite any existing template in the database. This parameter can only be used with */cfg*.
- *areas area1 area2*—Defines the areas of the security template to be used in the action against the database. Valid area names are: SECURITYPOLICY, GROUP_MGMT, USER_RIGHTS, REGKEYS, FILESTORE, and SERVICES. Multiple area names should be separated by a space. If no areas are specified, all areas are used.
- *export*—Used to create a security template from the current configuration of a system.
- *mergedpolicy*—Causes the export function to merge and export local and domain policy settings.
- *validate*—Used to verify the syntax of a security template before it is used.
- *FileName*—Defines the path and filename of the security template to validate.

AUDITING

Auditing is the security process that records the occurrence of specific operating system **events** in a Security Log. Every object in the Windows XP system has audit events related to it. These events can be recorded on a success or failure basis and in some cases based on users or groups. For example, logging all failed logon attempts may warn you when an attack that will breach your security is occurring, or monitoring classified documents for read access can let you know who is accessing them and when. Auditing can provide valuable information about security breaches, resource activity, and user adeptness. Auditing is also useful for investigating performance and planning for expansion.

Auditing is enabled through the Local Security Policy or through a domain policy (see Chapter 5). Once enabled, the audited events are recorded in the Security Log in Event Viewer. **Event Viewer** is accessed through the Administrative Tools (accessed from the Start menu or Control Panel) and maintains logs about application, security, and system events on your computer, enabling you to view and manage the event logs, gather information about hardware and software problems, and monitor Windows XP security events. To view the items related to auditing, select the Security Log node (see Figure 6-4). Double-clicking an event opens the Explain dialog box (see Figure 6-5). This particular audit event records the data about a successful logon of the Administrator account on the workstation named WXPPRO-102. Audit entries in the Security Log contain information about the event including user logon identification, the computer used, time, date, and the action or event that instigated an audit.

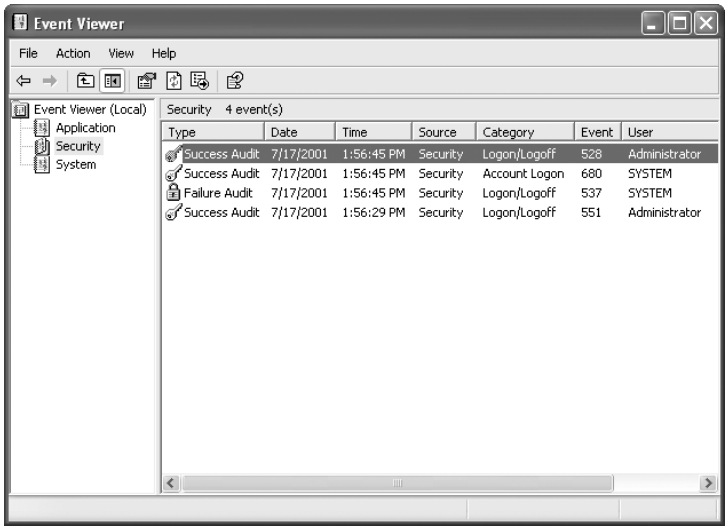

Figure 6-4 The Security Log viewed through the Event Viewer

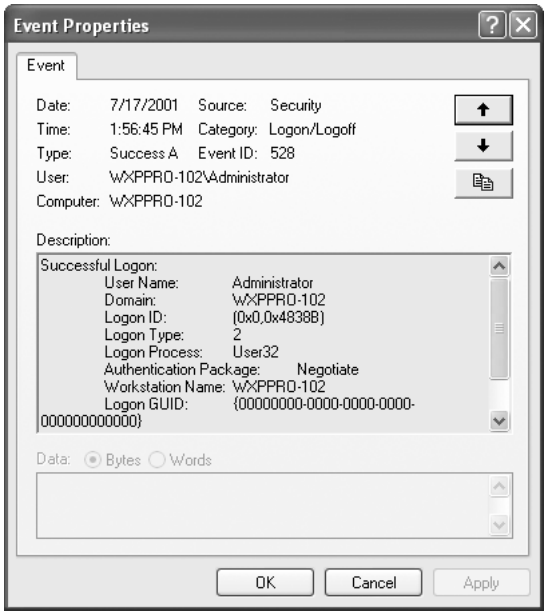

Figure 6-5 The Security Log event detail

If you select to audit object access on either a Success or Failure basis, you can define the actions or activities to audit for objects on an object-by-object basis for each possible action based on that object's type for specific users and groups. For example, you

might audit access to certain network resources such as files or printers by different users and/or groups. To set an object's auditing controls, perform the following steps:

Auditing can be configured only for users and resources that belong to a domain, not a work group.

1. Open the Properties dialog box for an NTFS object (such as a file, folder, or printer). Right-click the object, then select **Properties** from the pop-up menu.
2. Select the **Security** tab.
3. Click the **Advanced** button.
4. Select the **Auditing** tab. This displays all of the currently defined audit events for this object. It is blank by default.
5. Click the **Add** button.
6. Click the **Advanced** button.
7. Click the **Find Now** button.
8. Select a computer, group, or a user from the Select User, Computer, or Group dialog box.
9. Click **OK**.
10. Click **OK**.
11. Select either **Successful** or **Failed** for any of the listed actions for this object type. The selections made here are the actions that are recorded in the Security Log.

If you selected to record only Failures in the Local Security Policy, selecting Successful actions on this dialog box does not record items in the Security Log.

12. Click **OK**.
13. Repeat steps 5 through 12 for all users, computers, or groups.
14. Repeat steps 1 through 13 for all objects.

Auditing numerous objects or events can result in a large Security Log and can slow down network or computer performance.

The Event Viewer can be configured to monitor the size of the Security Log and to take action when it reaches a target size. The actions are Overwrite events as needed, Overwrite events older than XX days, or Do not overwrite events. If the maximum size is reached and Do not overwrite events is selected, an alert appears stating that the log needs to be cleared. To access these controls, select Properties from the menu that appears when right-clicking over the Security Log node in the Event Viewer.

Encrypted File System

Microsoft has extended the native NTFS file system to include encrypted storage. This new security measure, the **Encrypted File System (EFS)**, allows you to encrypt data stored on NTFS drive. When EFS is enabled on a file, folder, or drive, only the enabling user can gain access to the encrypted object. EFS is enabled through a checkbox accessed through the Advanced button on the General tab of an object's Properties dialog box.

EFS uses a public and private key encryption method. The private key is assigned to a single user account. No other user, computer, or operating system can gain access to the encrypted files. For the authorized user (i.e., the user with the correct private key), access to the encrypted files is unhindered. In fact, the entire encryption process is invisible to the user.

Encryption is just another attribute of NTFS; therefore, you should treat encryption in the same manner as attributes and permissions. Any new file created or copied into an encrypted folder assumes the settings of that folder. Moving an encrypted file to a nonencrypted folder allows the file to retain its original settings, but copying the encrypted file causes the file to assume the settings of the destination folder. Because EFS is an additional level of processing required by the operating system to grant access to file-level objects, the performance of the file system can be noticeably impaired. You'll need to perform your own baseline comparison of your storage system's performance to determine exactly how much degradation is caused by EFS.

If the encryption key is lost or the user account is deleted, there is a mechanism to recover encrypted files. This mechanism is called the recovery agent, which is defined through Group Policy under the Public Key Policies. EFS does not function without a recovery agent. In fact, Windows XP (and Windows 2000) automatically designates the local Administrator as the recovery agent until you specifically define another recovery agent. The recovery agent is able to decrypt files by logging onto the system where the files are stored and de-selecting the Encrypt checkbox on the files and folder's Advanced Properties dialog box. For more details on the Recovery agent, please consult the *Microsoft Windows XP Professional Resource Kit*.

Windows XP includes a command-line tool for batch-processing of encryption (i.e., encrypting or decrypting large numbers of files or folders through a command line or batch file). The CIPHER command has the following syntax:

```
CIPHER [/E|/D] [/S[:directory]] [/A] [/I] [/F] [/Q]
       [/H] [pathname [...]]
CIPHER /K
CIPHER /R:pathname
CIPHER /U [/N]
CIPHER /W:directory
```

The following list defines each of the CIPHER command's parameters:

- /A—Forces operation on files and folders.
- */D*—Decrypts the listed filename(s).
- */E*—Encrypts the listed filename(s).
- */F*—Forces encryption, even on already encrypted files.
- */H*—Shows files with the hidden or system attributed set.
- */I*—Ignores errors and proceeds with processing.
- */K*—Creates a new encryption key for the user.
- */N*—Prevents encryption keys from being updated. Must be used with /U.
- */Q*—Silences activity except for essential feedback.
- */R*—Generates a recovery agent key and certificate into .PFX and .CER files.
- */S*—Performs the action on all subcontents.
- */U*—Scans for all encrypted files on local drives to update user's encryption key.
- */W*—Removes deleted data in the available disk space.
- *Pathname*—Specifies a pattern, file, or directory. Wildcards can be used; each pattern must be separated by a space.
- *Directory*—Specifies a directory.

When CIPHER is used with only a filename and without parameters, the status of the object is displayed, indicating whether the object is encrypted and whether new files added to a folder will be encrypted.

The primary benefit of EFS is that if your computer is either physically accessed or stolen, the data is protected as long as the malicious user does not gain access to the username and password that holds the private key for the encrypted files. The primary drawback is the increased processing power required to encrypt all writes and decrypt all reads on the fly. This process negatively affects performance to a noticeable extent on many systems.

Internet Security

Connecting to the Internet requires that you accept some risk. That risk includes downloading Trojan horses or viruses, accepting malicious e-mail, or even allowing a remote cracker to take complete control of your computer. Most of the security features used to protect data within a LAN or even on a standalone system can also be leveraged to protect against Internet attacks. Plus, Microsoft has added the Internet Connection Firewall to Windows XP. The Internet Connection Firewall (ICF) is a simple firewall used to protect any network connection, especially dial-up or dedicated Internet links. ICF is discussed in detail in Chapter 8, "Internetworking with Remote Access."

Chapter Summary

- Windows XP has object-level access controls that provide the foundation on which all resource access rest. By comparing the access control lists associated with individual objects to the access tokens that define the rights of any user process, Windows XP decides which object access requests to grant and which to deny.
- The Windows XP logon process (WinLogon) strictly controls how users identify themselves and log onto a Windows XP machine. The attention sequence (Ctrl+Alt+Delete) prevents an unauthorized user from obtaining system access to domain clients or properly configured standalone clients. Likewise, WinLogon's protected memory structures keep this all-important gatekeeper function from being replaced by would-be system crackers. Authentication can take place using various encryption schemes, including Kerberos, SSL, or NTLM.
- WinLogon also supports a number of logon controls: handling of a default logon name, providing security notices, changing the default shell, handling system shutdown options, and enabling automatic logon. Key Local Computer Policy settings can be used to block unauthorized break-in attempts.
- The local computer policy controls many aspects of the security system as well as enabling or restricting specific functions and features of the operating system. You can use Windows XP auditing capabilities to track down errant behavior or detect when system problems may be occurring. Encrypted File System (EFS) protects your data with an encryption system. All in all, Windows XP offers a reasonably secure operating environment that is designed to help administrators keep their important assets safe from harm and unwanted exposure.

Key Terms

access control list (ACL) — A list of security identifiers that are contained by a resource object. Only those processes with the appropriate access token can activate the services of that object.

access token — Objects containing the security identifier of an active process. These tokens determine the security context of the process.

auditing — This is the process of tracking events by recording selected types of events in the Security Log.

authentication — The process of validating a user's credentials to allow access to certain resources.

certificate — An electronic identity verification mechanism. Certificates are assigned to a client or server by a Certificate Authority. When communications begin, each side of the transmission can decide to either trust the other party based on their certificate and continue the communications or not to trust and terminate communications.

domain — A collection of computers with centrally managed security and activities.

domain controller — A specified computer role of WindowsNT, 2000, or .NET Servers that authenticates domain logons and maintains the security policies and the account database for a domain.

domain security — The control of user accounts, group memberships, and resource access for all members of a network instead of for only a single computer.

Encrypted File System (EFS) — A security feature of NTFS under Windows XP that allows files, folders, or entire drives to be encrypted. Once encrypted, only the user account that enabled the encryption has the proper private key to decrypt and access the secured objects.

event — Any significant occurrence in the system or in an application that requires users to be notified or a log entry to be added. Types of events include audits, driver failures, user logon, process launching, system shutdown, etc.

Event Viewer — The utility that maintains application, security, and system event logs on your computer, enabling you to view and manage the event logs, gather information about hardware and software problems, and monitor Windows XP security events.

identification — The process of establishing a valid account identity on a Windows XP machine by supplying a correct and working domain name (if necessary) and account name.

IPSec (IP Security) — An encrypted communication mechanism for TCP/IP to create protected communication sessions. IPSec is a suite of cryptography-based protection services and security protocols.

Kerberos version 5 — An authentication encryption protocol employed by Windows XP to protect logon credentials.

Local Computer Policy — A Windows XP security control feature used to define and regulate security-related features and functions.

network authentication — The act of connecting to or accessing resources from some other member of the domain network. Network authentication is used to prove that you are a valid member of the domain, that your user account is properly authenticated, and that you have access permissions to perform the requested action.

NTLM (NT LAN Manager) authentication — The authentication mechanism used on Windows NT that is retained by Windows XP for backward compatibility.

object — Everything within the Windows XP operating environment is an object. Objects include files, folders, shares, printers, processes, etc.

password — A unique string of characters that must be provided before a logon or an access is authorized. Passwords are a security measure used to restrict initial access to Windows XP resources.

process — The primary unit of execution in the Windows XP operating system environment. A process may contain one or more execution threads, all associated with a named user account, SID, and access token. Processes essentially define the container within which individual applications and commands execute under Windows XP.

public key policy — A security control of Windows XP where recovery agents for EFS and domain-wide and trusted certificate authorities are defined and configured. These policies can be enforced on a user by user basis.

Secure Socket Layer/Transport Layer Security (SSL/TLS) — A mechanism used primarily over HTTP communications to create an encrypted session link through the exchange of certificates and public encryption keys.

security ID (SID) — A unique number that identifies a logged-on user to the security system. SIDs can identify one user or a group of users.

shell — The default user process that is launched when a valid account name and password combination is authenticated by the WinLogon process for Windows XP. The default shell of Windows XP is Windows Explorer. The default shell process manages the desktop, Start menu, taskbar, and other interface controls. The shell process defines a logged on user's runtime environment from this point forward, and supplies all spawned processes or commands with its access token to define their access permissions until that account logs out.

user account — This entity contains all of the information that defines a user to the Windows XP environment.

WinLogon — The process used by Windows XP to control user authentication and manage the logon process. WinLogon produces the logon dialog box where user name, password, and domain are selected, controls automated logon, warning text, the display of the shutdown button, and the display of the last user to log onto the system.

Review Questions

1. Which of the following should be used to define IPSec policies for a domain?
   a. TCP/IP Properties
   b. Local Computer Policy
   c. Event Viewer
   d. Group Policy
2. All processes in Windows XP require an access token. True or False?
3. A SID is a unique number and is never duplicated. True or False?
4. Permissions that are changed while the user is actively logged on do not take effect until that user logs on to the system again. True or False?

5. The default Windows XP authentication method is to supply valid domain and account names, plus a valid password; however, Windows XP permits use of alternate authentication techniques. True or False?
6. What is the first thing the security system looks for when it scans an ACL for an object?
   a. A Deny to the object for the requested service, at which point access is immediately denied
   b. Any Allow permission that provides the requested permission
   c. It checks the default, and if access is permitted thereby, allows the request to proceed
   d. None of the above
7. What is the default access level that Windows XP assigns to new objects by default?
   a. Restrict
   b. Allow
8. Which of the following is a good reason for adding DontDisplayLastUserName to the Windows XP Registry? (Choose all that apply.)
   a. To prevent easy discovery of user account names
   b. To improve security on a shared machine
   c. To reduce burnout on the machine's monitor
   d. To force users to provide a valid user name in addition to a password to logon
9. The Windows XP authentication process can be automated by adding default user information and the _____ value to the Registry.
   a. DontDisplayLastUsername
   b. AutoAdminLogon
   c. Legal Notice Caption
   d. AutomateLogon
10. Which of the following is the most likely reason for a security notice that appears when users attempt to logon to a Windows XP machine at the National Security Agency?
    a. To make sure that outsiders don't try to break into the system.
    b. To inform unauthorized users that they are subject to legal action if they obtain unauthorized access to the system.
    c. To remind valid system users about Acceptable Use Policies.
    d. None of the above

11. The default shell process for Windows XP is called the:
    a. Windows Explorer
    b. Program Manager
    c. Command shell
    d. C shell
12. The _____ is created by the Windows XP security subsystem at logon and identifies the current user to the subsystem.
    a. Access ID
    b. Security ID
    c. Group ID
    d. access token
13. The _____ key sequence initiates the classic logon process.
    a. Ctrl+Esc
    b. Alt+Tab
    c. Ctrl+Break
    d. Ctrl+Alt+Delete
14. An access token is required to access any Windows XP object. True or False?
15. To customize the security structure of your Windows XP system, you can change the behavior of the logon process. True or False?
16. What is the primary protocol that Windows XP uses for authentication?
    a. NTLM
    b. Secure Socket Layer
    c. Kerberos
    d. NetBIOS
17. Which of the following statements are true about the Local Computer Policy? (Choose all that apply.)
    a. It is used to control aspects of the Windows XP security system.
    b. It is used to assign user accounts to groups.
    c. It can be customized by third-party applications.
    d. It can be superceded by a domain's Group Policy.
18. What is the special-purpose application invoked by the Windows XP attention sequence that serves as the logon process?
    a. WINPOPUP
    b. WINLOGON
    c. USERMGR
    d. EXPLORER

19. What security feature is included in Windows XP specifically to protect TCP/IP communications between two systems?
    a. Kerberos version 5
    b. IPSec
    c. Strong passwords
    d. EFS
20. What is EFS used to protect?
    a. Passwords
    b. Data files
    c. Group Policy
    d. Communication sessions
21. If the Windows Explorer shell is replaced with the Program Manager shell, which of the following side effects will occur? (Choose all that apply.)
    a. No access to the Start menu
    b. No task bar
    c. No access to the Task Manager
    d. No more DOS Command prompt
22. Only the user who encrypted a file through EFS can access that file later. True or False?
23. What predefined IPSec policy should you use to employ encryption only when required by a remote system?
    a. Client (Respond Only)
    b. Server (Request Security)
    c. Secure Server (Require Security)
24. Auditing can be defined for an object for specific users and groups for one or more individual services or actions. True or False?
25. Audit events are recorded in the System log. True or False?

HANDS-ON PROJECTS

Project 6-1

To open the Local Computer Policy:

1. Open the **Run** command (**Start**|**Run**).
2. Type **mmc**, then click **OK**. This launches the Microsoft Management Console.
3. Select **Add/Remove Snap-in** from the File menu.

4. Click the **Add** button.
5. Locate and select **Group Policy**.
6. Click **Add**.
7. On the Select Group Policy object dialog box, notice that Local Computer is listed by default. Click **Finish**.
8. Click **Close** on the Add Standalone Snap-in dialog box.
9. Click **OK** on the Add/Remove Snap-in dialog box.
10. The Local Computer Policy node should now appear in the MMC.

Project 6-2

To disable the display of the last user name on the logon screen:

This hands-on project requires that you first complete Hands-on Project 6-1.

1. In the Local Computer Policy console, click the **boxed plus sign** beside Local Computer Policy to expand its contents.
2. Locate the **Computer Configuration** node. Click its **boxed plus sign** to expand its contents.
3. Locate the **Windows Settings** node. Click on its **boxed plus sign** to expand its contents.
4. Locate the **Security Settings** node. Click on its **boxed plus sign** to expand its contents.
5. Locate the **Local Policies** node. Click on its **boxed plus sign** to expand its contents.
6. Locate and select the **Security Options** node.
7. In the Details pane, locate and select **Interactive logon: Do not display last user name**.
8. Select the **Action** menu, then click **Properties**. The Local Security Policy Setting dialog box for the selected control is displayed.
9. Select the **Enable** radio button.
10. Click **OK**.
11. Log off and log back in. Notice that the last logged on user name is no longer displayed.

Project 6-3

To display a legal warning message at logon:

This hands-on project requires that you first complete Hands-on Project 6-1.

1. In the Local Computer Policy console, locate and select the following subnode: **Computer Configuration | Windows Settings | Security Settings | Local Policies | Security Options**.
2. Locate and select **Interactive logon: Message title for users attempting to log on**.
3. Select the **Action** menu, then click **Properties**.
4. Type **Warning!** in the field. Click **OK**.
5. Select **Interactive logon: Message text for users attempting to log on**.
6. Select the **Action** menu, then click **Properties**.
7. In the field type a warning message similar to the following: (Note: This excellent security warning message is reproduced from *The Windows NT Security Handbook*, by Tom Sheldon, Osborne/McGraw-Hill: Berkeley, 1997) **"Authorized Users Only! The information on this computer and network is the property of [name organization here] and is protected by intellectual property law. You must have legitimate access to an assigned account on this computer to access any information. You are permitted only to access information as defined by the system administrators. Your activities may be monitored. Any unauthorized access will be punished to the full extent of the law."**
8. Click **OK**.
9. Log off then log back in to see the warning message displayed between pressing Ctrl+Alt+Delete and the display of the WinLogon dialog box.

Project 6-4

To change the default shell:

Changing the shell will result in a new user interface. The Program Manager does not offer a Start menu, taskbar, Task Manager, and many other interface controls you are accustomed to from Windows XP. Employ this hands-on project with caution.

1. Open the **Run** command (**Start | Run**).
2. Type **regedit**. Click **OK**.

3. Locate and select the key: **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.**
4. Locate and select the **Shell** value.
5. Select **Modify** from the Edit menu.
6. Change the value data from EXPLORER.EXE to **PROGMAN.EXE**.
7. Click **OK**.
8. Select **Exit** from the Registry menu.
9. The system is now configured to launch the Windows NT 3.51 Program Manager as the shell. To return the shell to Windows Explorer, either reopen the Registry editor now and change the Shell value back to its original setting (EXPLORER.EXE), or, if you have already logged in with Program Manager as the shell, use the Run command from the File menu of the Program Manager to launch Regedit and make the change.

Project 6-5

To encrypt a folder with EFS:

The folder must be on an NTFS file system to complete this exercise.

1. Launch Windows Explorer (**Start**|**All Programs**|**Accessories**|**Windows Explorer**).
2. Expand My Computer and select the **C** drive in the left column. (If drive C is not formatted with NTFS, select some other drive that is formatted with NTFS).
3. Select the **File** menu, then **New**, and select **Folder** from the submenu.
4. Type in a name for the folder (such as **EFStemp**), then press **Enter**.
5. Right-click the new folder, then select **Properties** from the menu.
6. On the **General** tab, click the **Advanced** button.
7. Click to place a check in the **Encrypt contents to secure data** checkbox.
8. Click **OK**.
9. Click **OK**.
10. Log off (**Start**|**Log Off**|**Log Off**)
11. Logon to the system with a different user account (**Ctrl+Alt+Delete**, then provide a different user name and password).
12. Launch Windows Explorer (**Start**|**All Programs**|**Accessories**|**Windows Explorer**).

13. Locate and try to access the EFStemp folder. Notice that you are unable to gain access.
14. Log off (**Start**|**Log Off**|**Log Off**)
15. Logon with the user account used to encrypt the folder (**Ctrl+Alt+Delete**, then provide user name and password).
16. Locate and try to access the **EFStemp** folder. Notice that you are able to gain access.
17. Right-click the **EFStemp** folder, then select **Properties** from the menu.
18. On the **General** tab, click the **Advanced** button.
19. Deselect (uncheck) the **Encrypt contents to secure data** checkbox.
20. Click **OK**.
21. Click **OK**.

Project 6-6

To explore the Local Computer Policy:

This hands-on project requires that you first complete Hands-on Project 6-1.

1. Expand the **Computer Configuration** node of the Local Computer Policy.
2. Expand the **Administrative Templates** node.
3. Expand each of the **Windows Components**, **System**, **Network**, and **Printers** subnodes.
4. Select each subnode one by one. Review the control details contained in each.
5. To open the Properties of a control detail, select it then select the Action menu, then click **Properties**.
6. View the Policy and Explain tabs of all control details that interest you.
7. Expand the **User Configuration** node and all of its subnodes.
8. Perform the same expansion and exploration as you did under the Computer Configuration node.
9. Select the **Exit** command from the Console menu of the MMC to close the utility. Click **Cancel** to discard any changes, if prompted.

Project 6-7

To set permissions on a file or folder:

This hands-on project requires that Windows XP be installed and an NTFS partition is present. Additionally, the computer must be a member of a domain.

1. Open Windows Explorer (**Start**|**All Programs**|**Accessories**|**Windows Explorer**).
2. In the left pane, select a drive formatted with NTFS within My Computer.
3. In the right pane, select a file or folder.
4. From the File menu, select **Properties**.
5. Select the **Security** tab.
6. Click **Add**.
7. Click the **Advanced** button.
8. Click the **Find Now** button.
9. Select the **Authenticated Users** group.
10. Click **OK**.
11. Click **OK**.
12. Select the **Authenticated Users** group, which now appears in the list of names on the Security tab for the NTFS object.
13. Select the **Modify** checkbox in the Allow column.
14. Select the **Everyone** group. Notice how the defined permissions for these two groups differ.
15. Click **OK**.

Project 6-8

To enable file access auditing:

1. Open the **Control Panel** (**Start**|**Control Panel**), and click **Switch to Classic View**.
2. Open the **Administrative Tools** by double-clicking on its icon.
3. Open the **Local Security Policy** by double-clicking on its icon.
4. Expand the **Local Policies** node by double-clicking on it.
5. Select the **Audit Policy** node.
6. Double-click the **Audit object access** item.
7. Select the **Success** checkbox. Click **OK**.

8. Launch Windows Explorer (**Start|All Programs|Accessories|Windows Explorer**).
9. Locate and select any text document on your computer, such as ***%system-root%*\Winnt\setuplog.txt**.
10. Select **File|Properties**.
11. Select the **Security** tab.
12. Click the **Advanced** button.
13. Select the **Audit** tab.
14. Click **Add**.
15. Click the **Advanced** button.
16. Click the **Find Now** button.
17. Select **Authenticated Users**.
18. Click **OK**.
19. Click **OK**.
20. Select the **List Folder/Read Data** checkbox under **Successful**.
21. Click **OK**.
22. Click **OK**.
23. Click **OK**.
24. Double-click the text file to open it.
25. Close Notepad by selecting **File|Exit**.
26. Return to **Administrative Tools** by clicking on its button on the taskbar.
27. Open the **Event Viewer** by double-clicking its icon.
28. Select the **Security** Log.
29. Double-click one of the event details.
30. Using the arrow buttons, scroll through the most recent event details to locate an event dealing with the successful reading of the text file.
31. Click **OK** to close the Event detail.
32. Close the Event Viewer by clicking the **X** button in the title bar.
33. Close Administrative Tools by selecting **File|Close**.
34. Close Windows Explorer by selecting **File|Close**.
35. On the Local Security Settings dialog box, double-click **Audit Object Access**.
36. Deselect **Success**.
37. Click **OK**.
38. Close the Local Security Settings dialog box by clicking the **X** button in the title bar.

CASE PROJECTS

1. You've been assigned the task of defining a security policy for your company. You've been given basic guidelines to follow. These include preventing users from installing software, securing the logon process, and enforcing disk quotas. Using the Local Computer Policy, detail the control you should configure and what settings you think would work best to accomplish these goals.
2. You've recently inherited the responsibility of administering a Windows XP network. The last administrator was rather lax in restricting user access. After working through the data folders to correct the access permissions, you suspect that some users still have access to confidential files. What can you do to determine if this type of access is still occurring? Describe the steps involved in enabling this mechanism and examining the results.